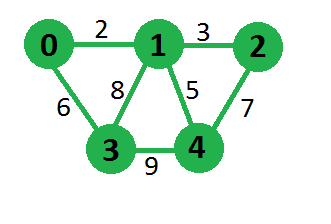
Minimum Product Spanning Tree

Last Updated: 08-07-2019

Given a connected and undirected graph, a spanning tree of that graph is a subgraph that is a tree and connects all the vertices together. A single graph can have many different spanning trees. A minimum product spanning tree for a weighted, connected and undirected graph is a spanning tree with weight product less than or equal to the weight product of every other spanning tree. The weight product of a spanning tree is the product of weights corresponding to each edge of the spanning tree. All weights of the given graph will be positive for simplicity.

Examples:

[[](https://media.geeksforgeeks.org/wp-content/uploads/graphForMST.png)](https://media.geeksforgeeks.org/wp-content/uploads/graphForMST.png)

Minimum Product that we can obtain is

180 for above graph by choosing edges

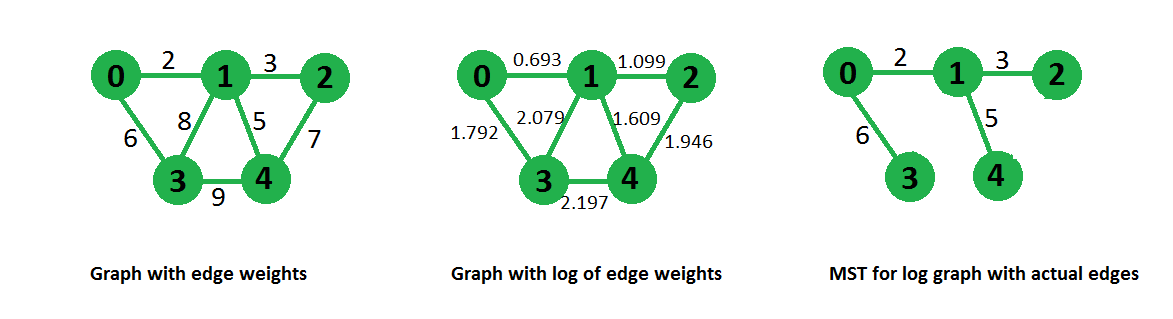
0-1, 1-2, 0-3 and 1-4

[**Recommended: Please try your approach on *{IDE}* first, before moving on to the solution.**](https://ide.geeksforgeeks.org/)

This problem can be solved using standard minimum spanning tree algorithms like [krushkal](https://www.geeksforgeeks.org/greedy-algorithms-set-2-kruskals-minimum-spanning-tree-mst/) and [prim](https://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-minimum-spanning-tree-mst-2/)’s algorithm, but we need to modify our graph to use these algorithms. Minimum spanning tree algorithms tries to minimize total sum of weights, here we need to minimize total product of weights. We can use property of [logarithms](https://en.wikipedia.org/wiki/Logarithm) to overcome this problem.  
As we know,

log(w1\* w2 \* w3 \* …. \* wN) =

log(w1) + log(w2) + log(w3) ….. + log(wN)

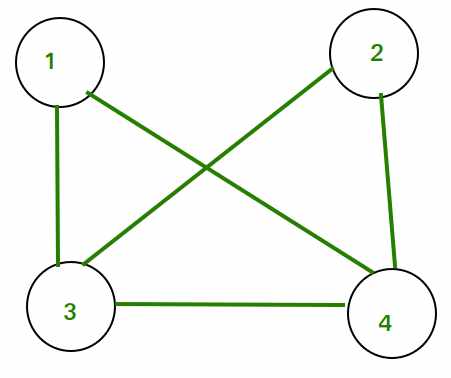
We can replace each weight of graph by its log value, then we apply any minimum spanning tree algorithm which will try to minimize sum of log(wi) which in-turn minimizes weight product.  
For example graph, steps are shown in below diagram,  
 [[](https://media.geeksforgeeks.org/wp-content/uploads/logGraph.png)](https://media.geeksforgeeks.org/wp-content/uploads/logGraph.png)

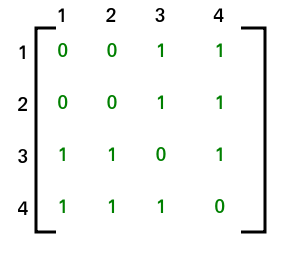
Total number of Spanning Trees in a Graph

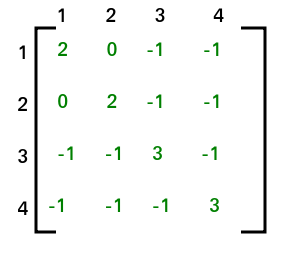
Last Updated: 17-05-2018

If a graph is a complete graph with n vertices, then total number of spanning trees is n(n-2) where n is the number of nodes in the graph. In complete graph, the task is equal to counting different labeled trees with n nodes for which have [Cayley’s formula](https://www.geeksforgeeks.org/g-fact-20-cayleys-formula-for-number-of-labelled-trees/).

**What if graph is not complete?**  
Follow the given procedure :-  
STEP 1: Create Adjacency Matrix for the given graph.  
STEP 2: Replace all the diagonal elements with the degree of nodes. For eg. element at (1,1) position of adjacency matrix will be replaced by the degree of node 1, element at (2,2) position of adjacency matrix will be replaced by the degree of node 2, and so on.  
STEP 3: Replace all non-diagonal 1’s with -1.  
STEP 4: Calculate co-factor for any element.  
STEP 5: The cofactor that you get is the total number of spanning tree for that graph.

Consider the following graph:  


Adjacency Matrix for the above graph will be as follows:  


After applying STEP 2 and STEP 3, adjacency matrix will look like  


The co-factor for (1, 1) is 8. Hence total no. of spanning tree that can be formed is 8.